Adjoint methods for 3D remote sensing

William Martin

Brian Cairns, Guillaume Bal

December 3, 2014

NASA Goddard Institute for Space Studies

Work supported by NASA Headquarters under the NASA Earth and Space Science Fellowship Program, Grant NNX-10AN85H, and in part by an IGERT grant from the US National Science Foundation to Columbia University.

Goddard Institute for Space Studies

Outline:

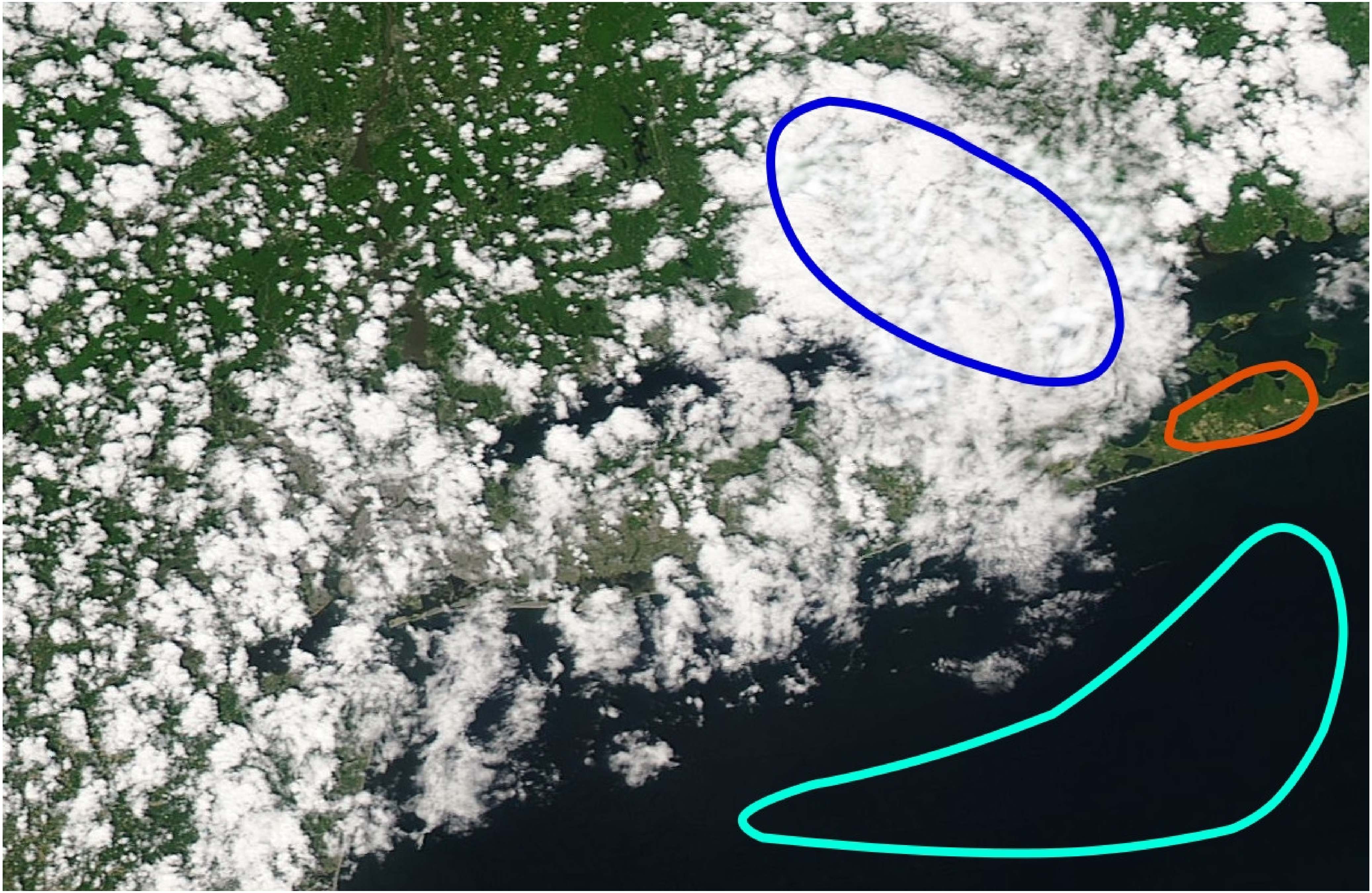
Adjoint methods for 3D remote sensing

- ***Why do we need 3D radiative transfer?***
- ***Why do we need adjoint methods?***
- ***What is an adjoint method?***

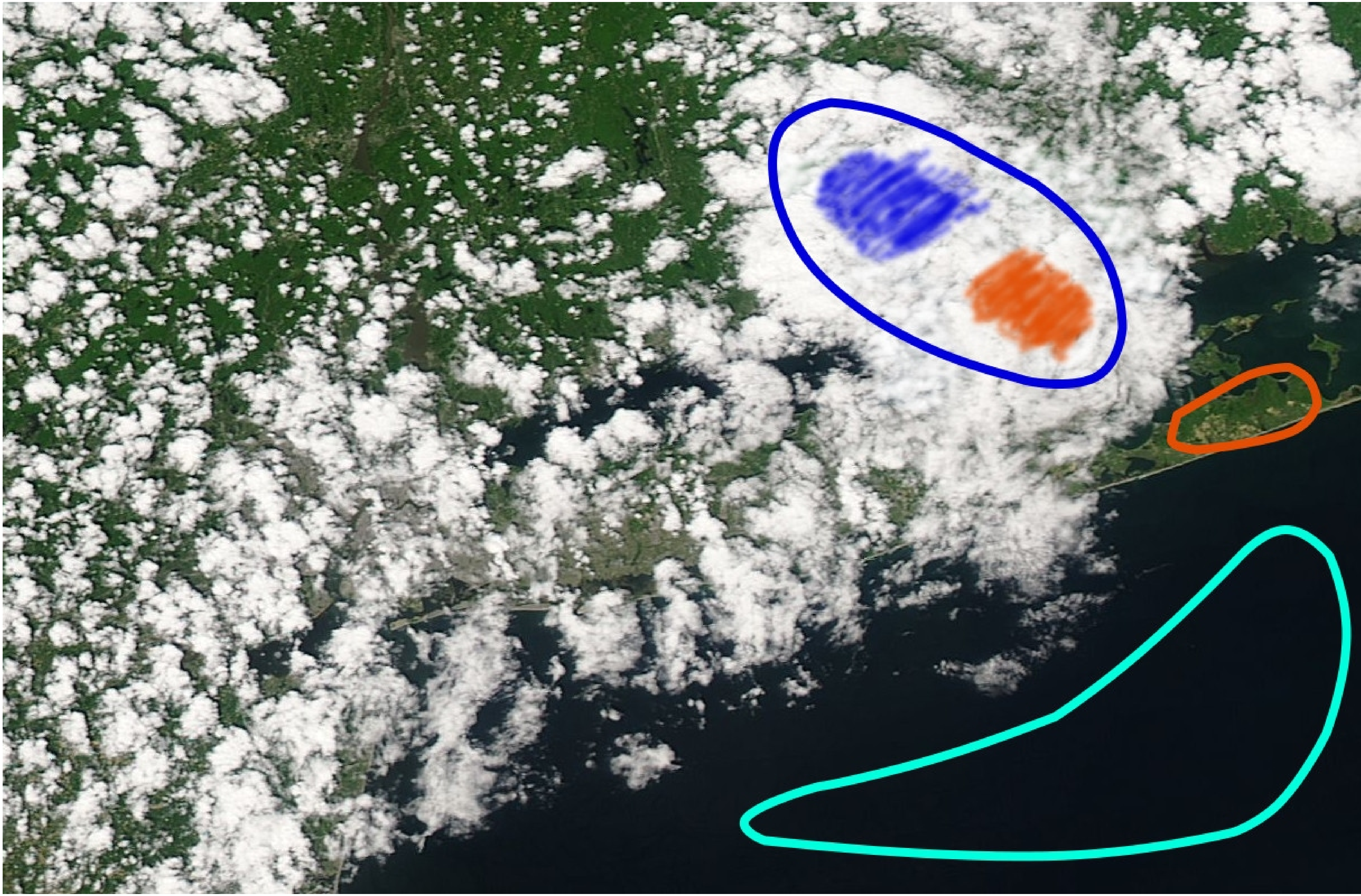
Outline:
Adjoint methods for 3D remote sensing

- ***Why do we need 3D radiative transfer?***
- ***Why do we need adjoint methods?***
- ***What is an adjoint method?***

Retrievals in 1D can work well

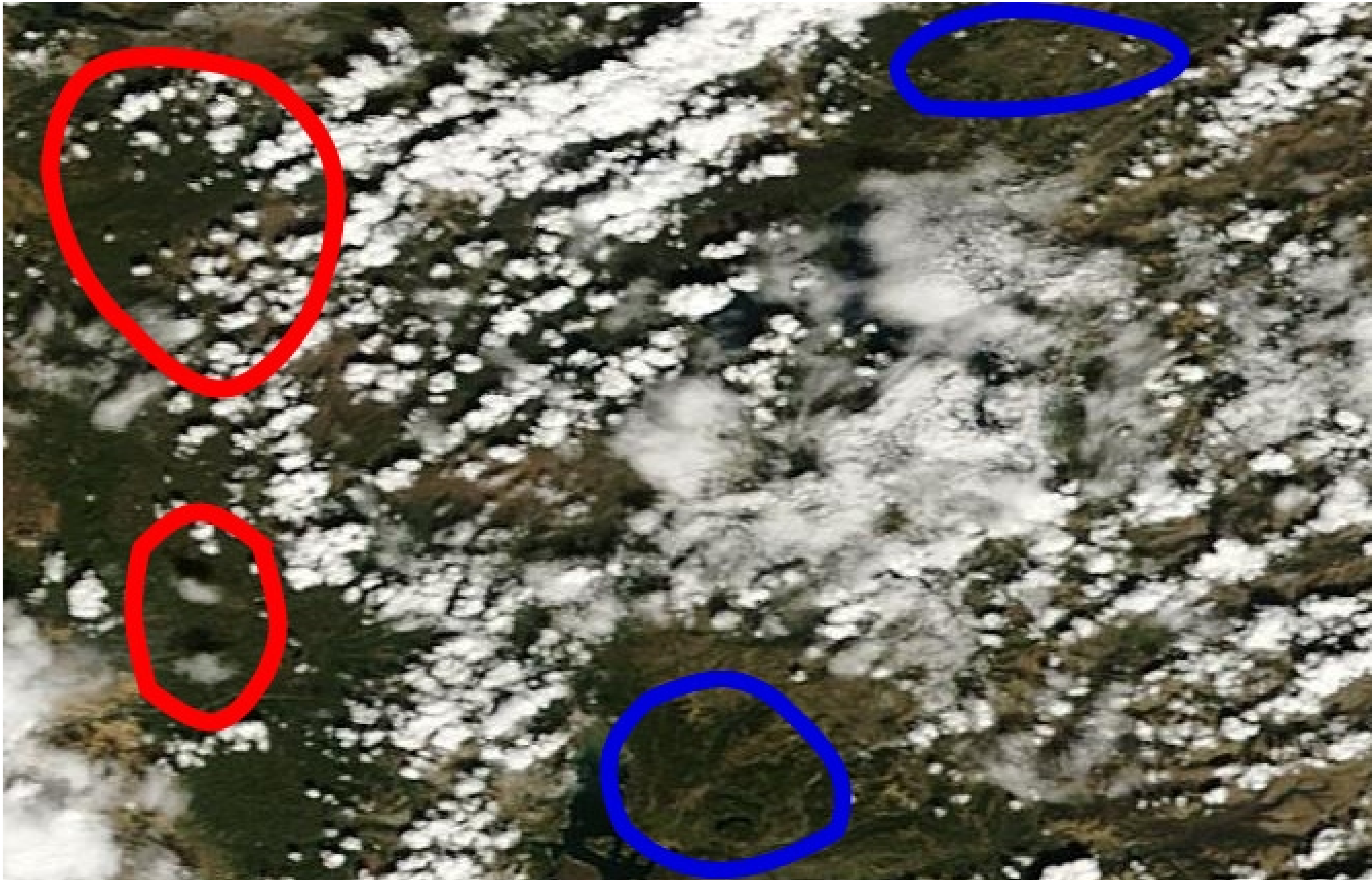


Retrievals in 1D can work well

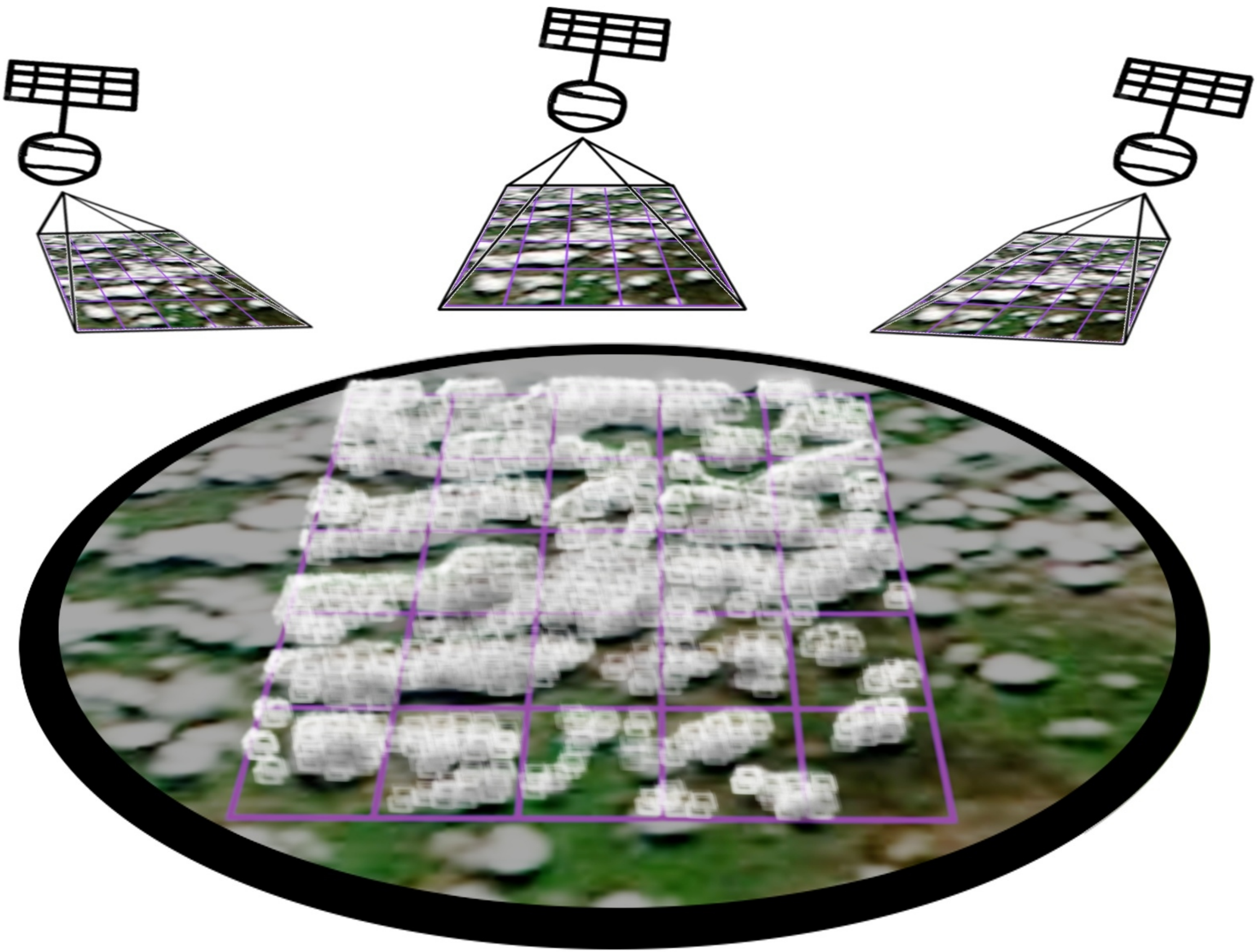


Retrievals in 3D can extend coverage

Retrievals in 3D can exploit 3D effects



The cost:
dependent pixels and 3D radiative transfer

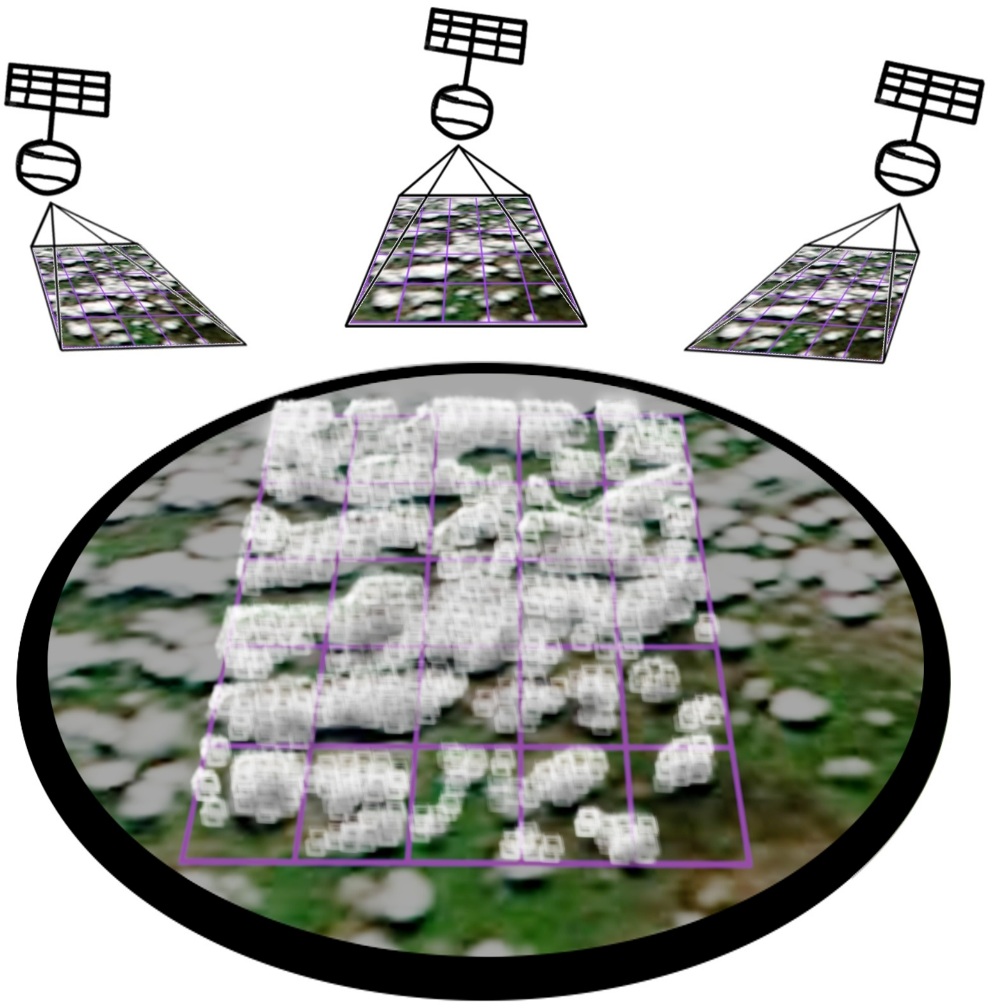


Outline:
Adjoint methods for 3D remote sensing

- *Why do we need 3D radiative transfer?*
- ***Why do we need adjoint methods?***
- *What is an adjoint method?*

Cloud representation:

Three-dimensional cloud



Consider 3D retrievals to extend coverage to broken cloud fields

Solver 3D VRTE

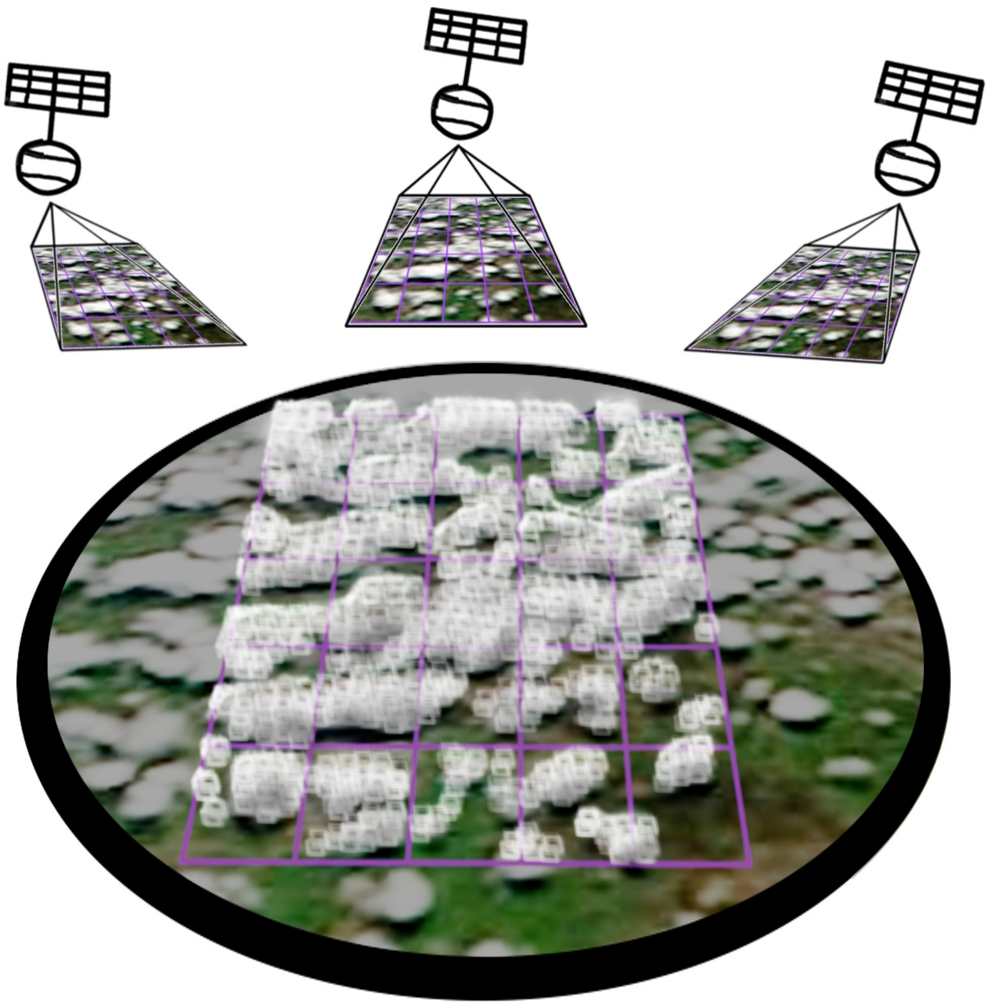
- SHDOM [Evans,1998 and 2014]
- Polarization [Doicu, Ef., Tr. 2013]
- Adjoint derivative [Martin, 2014]

Inverse problem

- Retrieval of 1D cloud properties [Evans, 2008]
- Stability and data requirements?

Cloud representation:

Three-dimensional cloud



Consider 3D retrievals to extend coverage to broken cloud fields

Solver 3D VRTE

- SHDOM [Evans,1998 and 2014]
- Adjoint derivative [Martin, 2014]

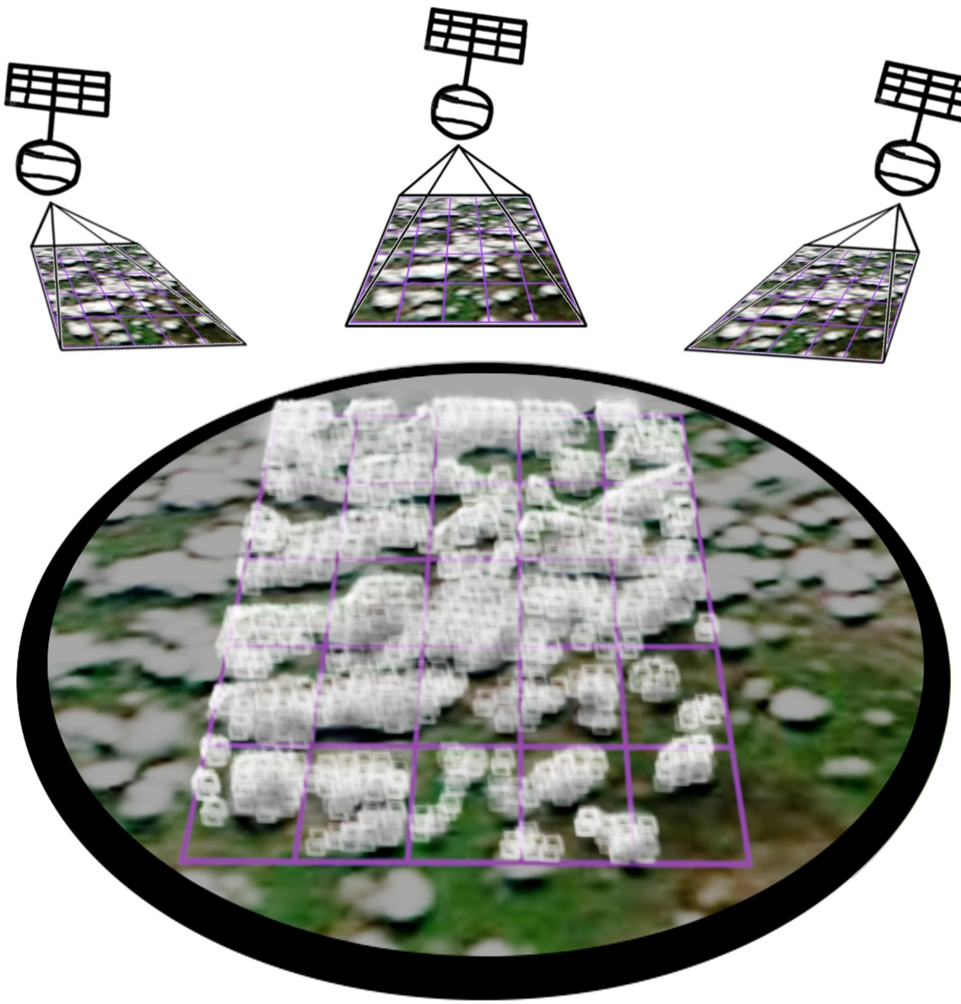
Inverse problem

- Retrieval of 1D cloud properties [Evans, 2008]
- Stability and data requirements?

How do we represent clouds for doing 3D retrievals of the atmosphere and surface?

Cloud representation:

Three-dimensional cloud



Measurements (future)

- Passive polarimetric imaging and active LIDAR and RADAR.

  ~100 images

  x

  ~10,000 pixels per image

  =

  ~1,000,000 total measurement constraints

Unknown parameters to retrieve

- *Cloud, aerosols and surface for each patch*

  ~1,000 volume and surface elements

  X

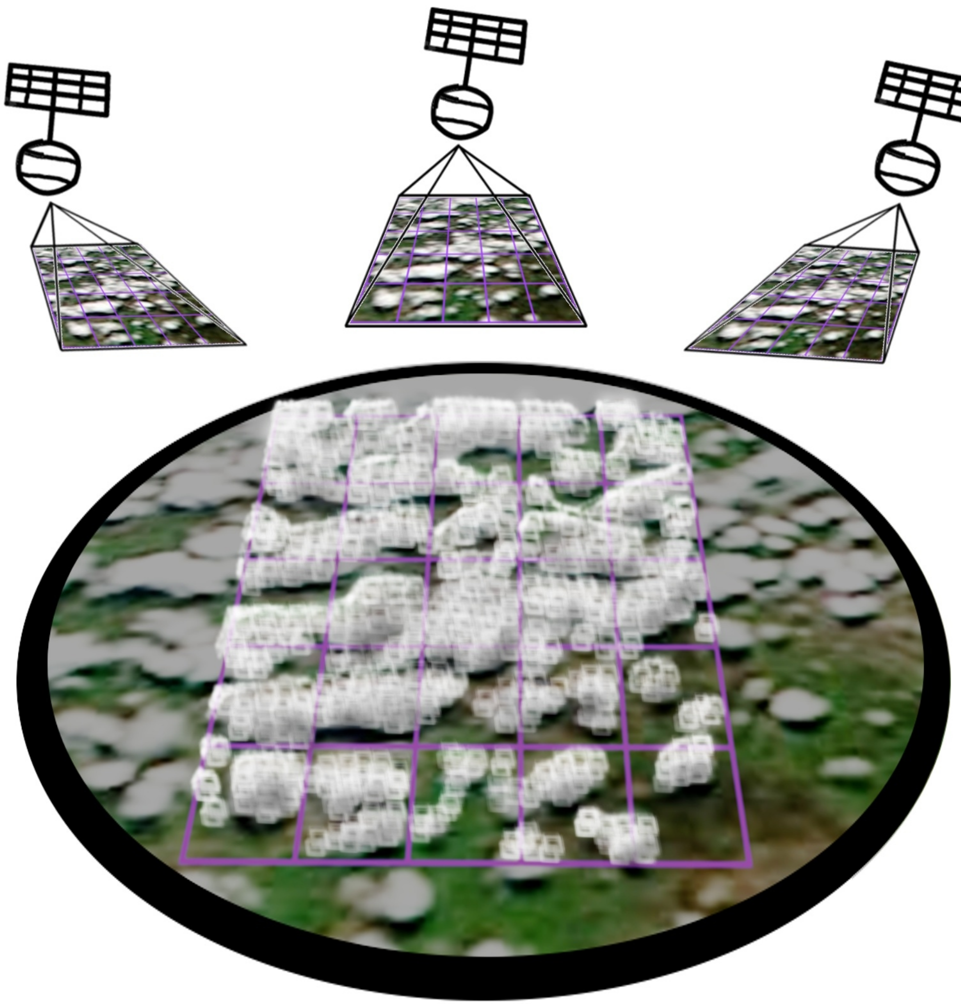
  ~10 properties (aerosol, cloud or surface)

  =

  ~10,000 total unknown parameters

Cloud representation:

Three-dimensional cloud



Measurements (future)

- Passive polarimetric imaging and active LIDAR and RADAR.

  ~100 images

  x

  ~10,000 pixels per image

  =

  ~1,000,000 total measurement constraints

Unknown parameters to retrieve

- *Cloud, aerosols and surface for each patch*

  ~1,000 volume and surface elements

  X

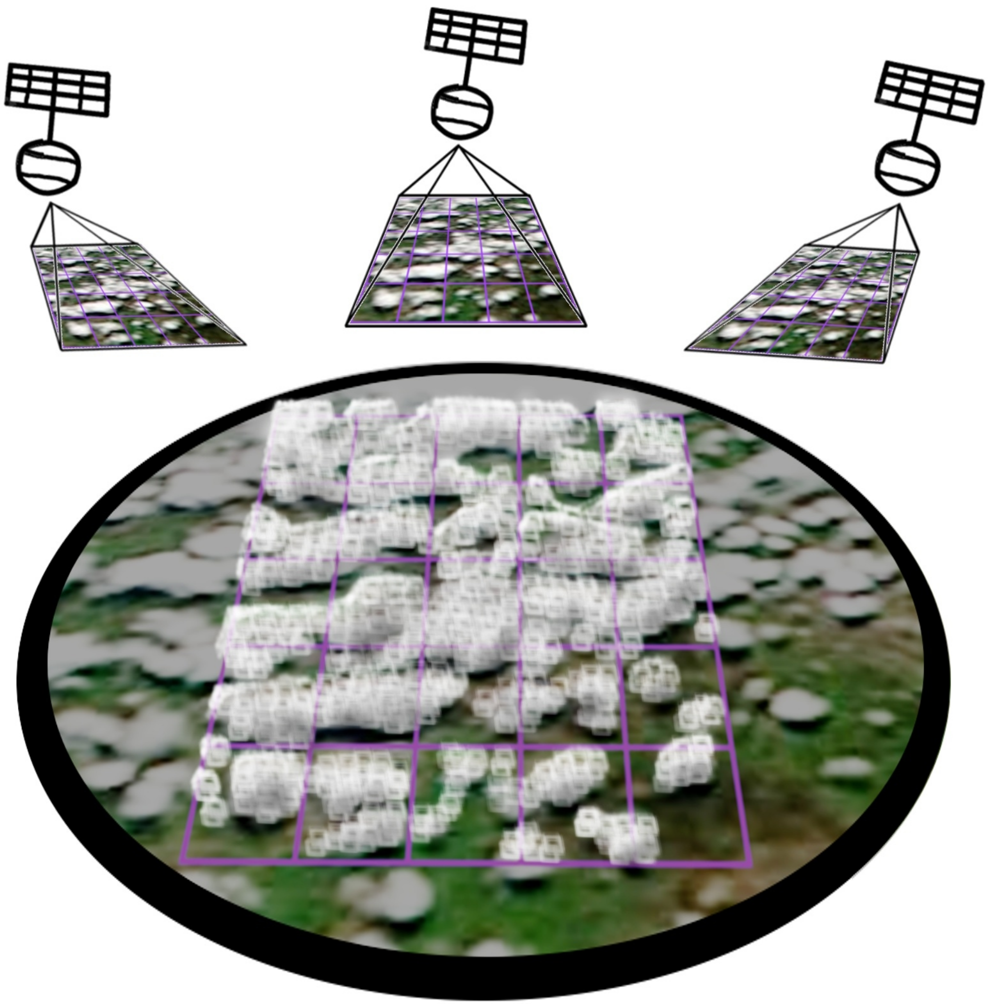
  ~10 properties (aerosol, cloud or surface)

  =

  ~10,000 total unknown parameters

This gives a large scale inverse problem:
Adjust 10,000 unknowns to fit 1,000,000 data.

Inverse problem:
find the cloud which fits data "best"



Minimize the misfit

- Non-linear least squares problem
- Solve by iterative methods

Requires the evaluation of the

$$\Phi\left(\boldsymbol{a}\right) = \frac{1}{2}\left(\hat{\boldsymbol{y}} - \boldsymbol{y}(\boldsymbol{a})\right)^T \cdot \boldsymbol{S}_{\boldsymbol{\epsilon}}^{-1} \cdot \left(\hat{\boldsymbol{y}} - \boldsymbol{y}(\boldsymbol{a})\right)$$

And the derivative of the misfit (steepest descent)

$$-\frac{\partial\Phi(\boldsymbol{a})}{\partial a^n} = \left(\hat{\boldsymbol{y}} - \boldsymbol{y}(\boldsymbol{a})\right)^T \cdot \left(\boldsymbol{S}_{\boldsymbol{\epsilon}}^{-1}\right) \cdot \frac{\partial \boldsymbol{y}(\boldsymbol{a})}{\partial a^n}$$

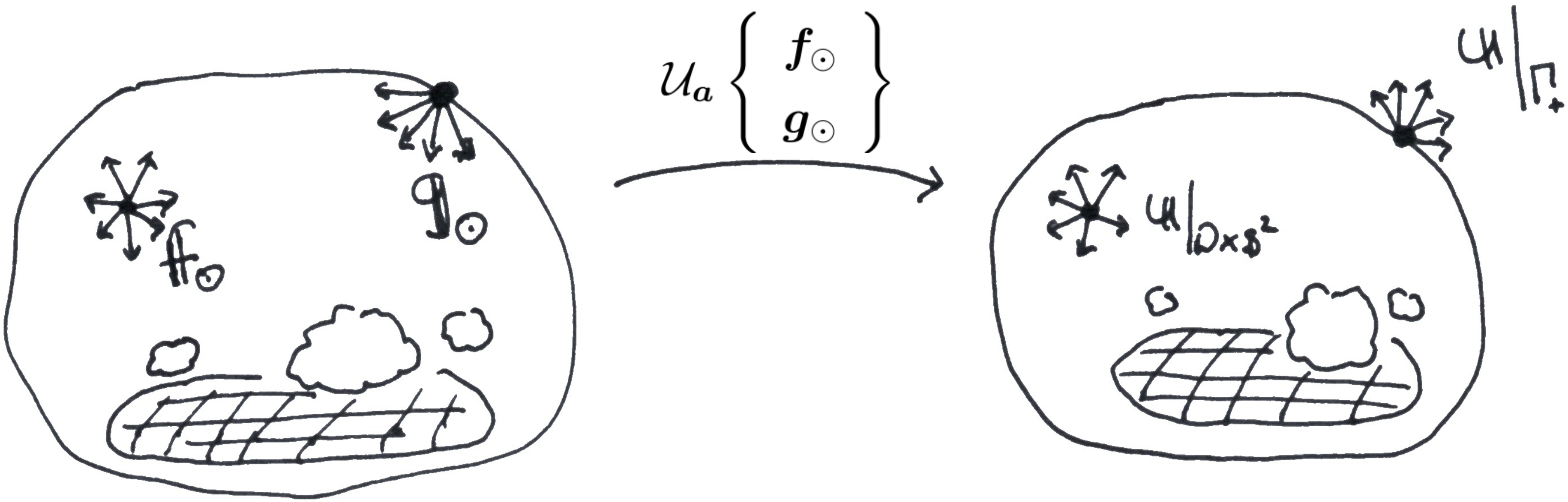
Outline:

Adjoint methods for 3D remote sensing

- *Why do we need 3D radiative transfer?*
- *Why do we need adjoint methods?*
- ***What is an adjoint method?***

Adjoint method:
to evaluate y(a) we need a solver.

For a fixed atmosphere and surface, the solver transforms volume-source and incoming-source functions into the internal and outgoing Stokes vectors.



Boundary value problem for 3D Radiative Transfer

$$\begin{aligned} \boldsymbol{v}\cdot\nabla\boldsymbol{u} + \sigma\boldsymbol{u} - \mathcal{Z}[\boldsymbol{u}] &= \boldsymbol{f}_\odot \quad \text{on} \quad D\times\mathbb{S}^2, \\ \boldsymbol{u}|_{\Gamma_-} - \mathcal{R}\big[\boldsymbol{u}|_{\Gamma_+}\big] &= \boldsymbol{g}_\odot \quad \text{on} \quad \Gamma_-. \end{aligned}$$

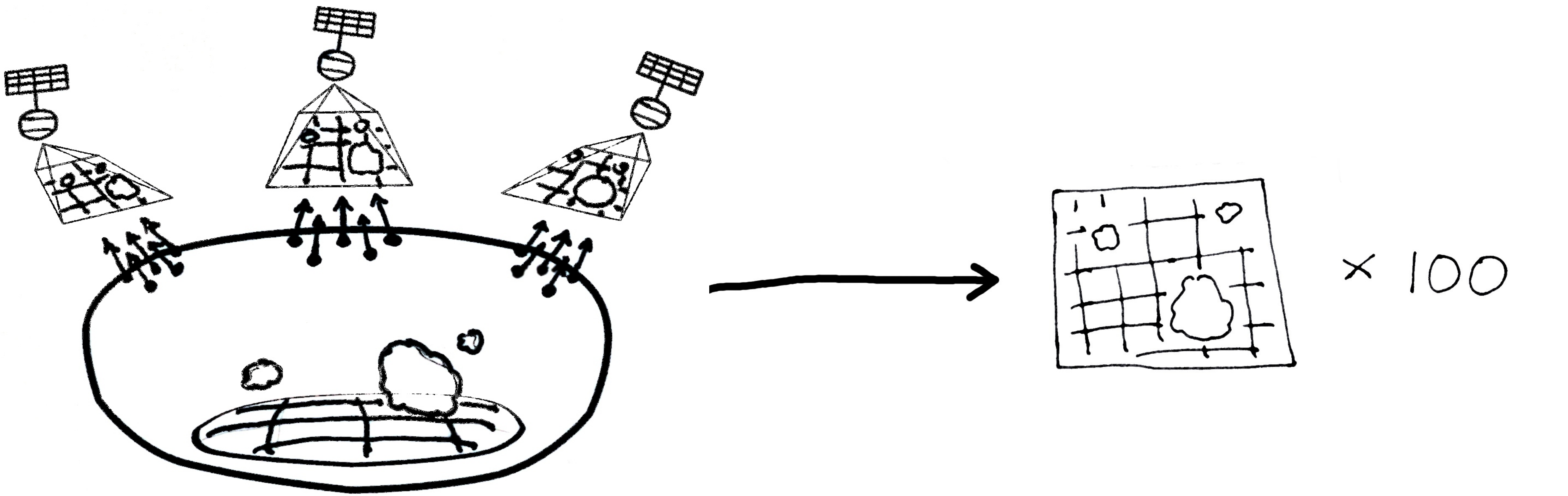
Solution operator (i.e. solver)

$$\left\{ \begin{array}{c} \boldsymbol{u}|_{D\times\mathbb{S}^2} \\ \boldsymbol{u}|_{\Gamma_+} \end{array} \right\} = \mathcal{U}_a \left\{ \begin{array}{c} \boldsymbol{f}_\odot \\ \boldsymbol{g}_\odot \end{array} \right\}$$

Adjoint method:

y(a) is a integral over the solution

Integrate the Stokes vector solution over the polarimetric response of each pixel.

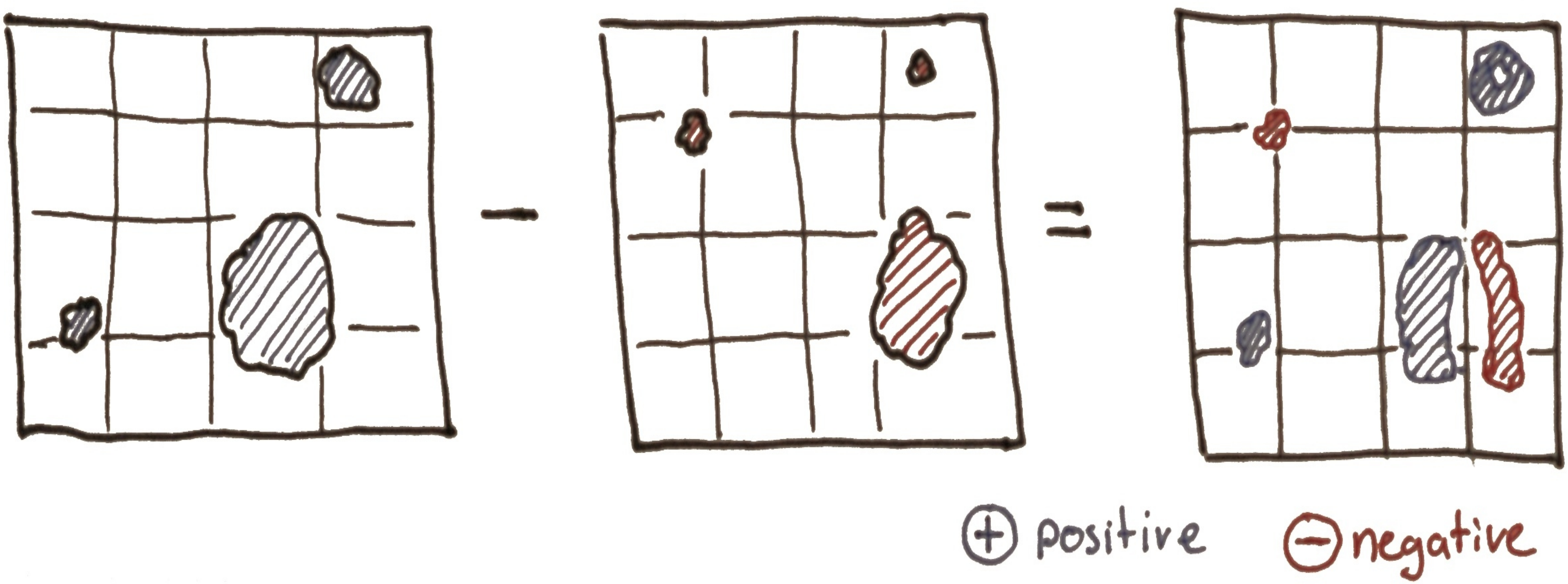


$$y^m(\boldsymbol{a}) = \left\langle \left\{ \begin{array}{c} \boldsymbol{p}_\odot^m \\ \boldsymbol{q}_\odot^m \end{array} \right\}, \mathcal{U}_{\boldsymbol{a}} \left\{ \begin{array}{c} \boldsymbol{f}_\odot \\ \boldsymbol{g}_\odot \end{array} \right\} \right\rangle_{D\times\mathbb{S}^2\oplus\Gamma_+}$$

Adjoint method:

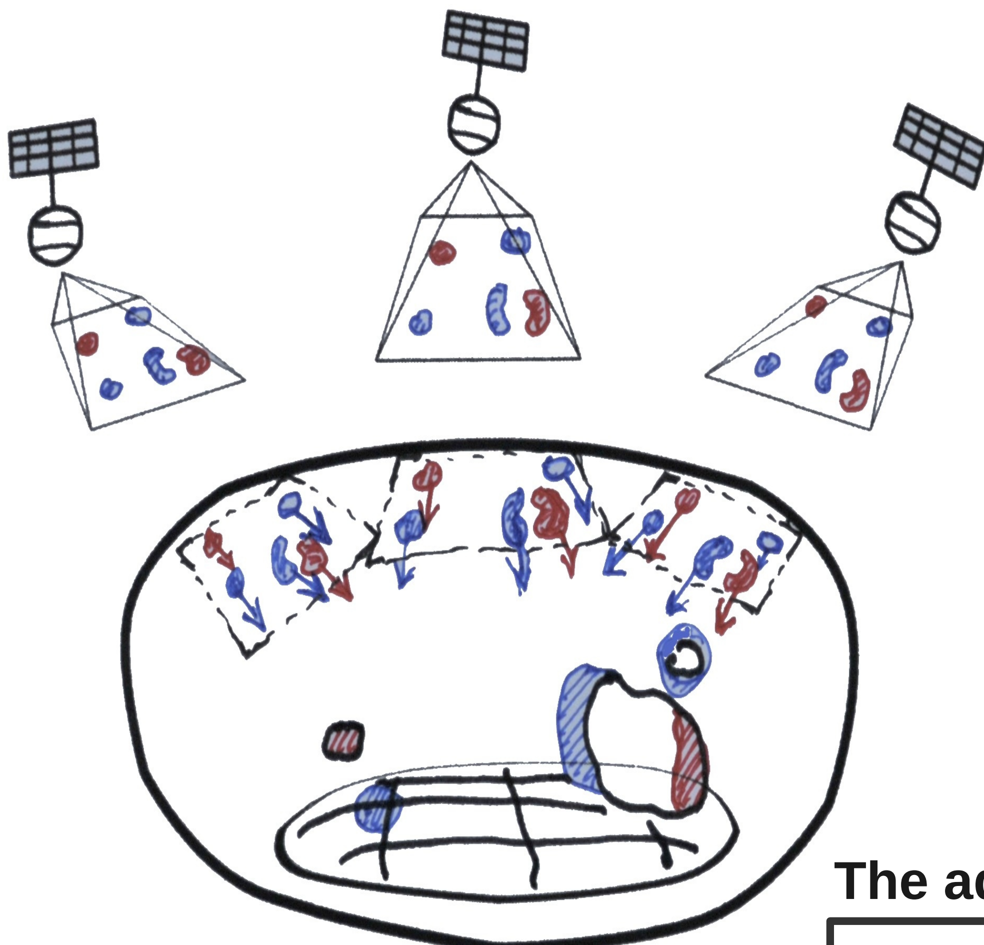
compute the measurement residual

y_{data} $\qquad$ $y(a)$



Adjoint method:

solve the adjoint 3D VRTE



The measurement residual is on the wrong domain

- 2D images in space

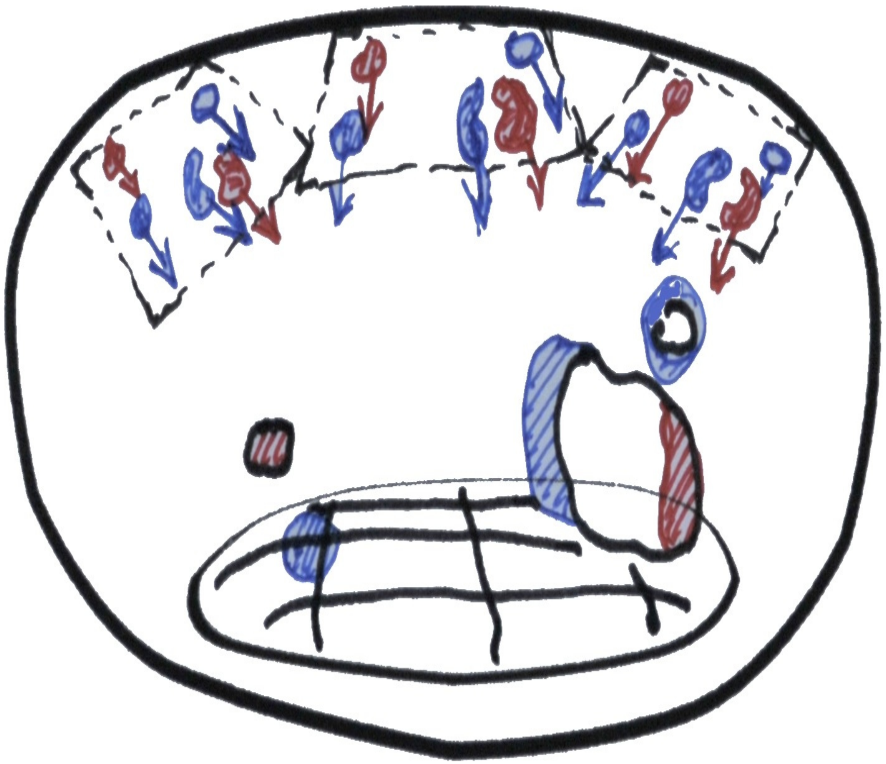
The adjoint solution is on the right domain

- 3D atmosphere and surface

The adjoint solver calls the forward solver

$$\boxed{\mathcal{U}_{\boldsymbol{a}}^{*}\left\{\begin{array}{c}\boldsymbol{p}_{\odot}\\ \boldsymbol{q}_{\odot}\end{array}\right\}=\alpha\boldsymbol{Q}\mathcal{U}_{\boldsymbol{a}}\left\{\begin{array}{c}\alpha\boldsymbol{Q}\boldsymbol{p}_{\odot}\\ \alpha\boldsymbol{Q}\boldsymbol{q}_{\odot}\end{array}\right\}}$$

Adjoint method:
compute the steepest descent of misfit



The adjoint Stokes-vector solution is on the correct 3D domain.

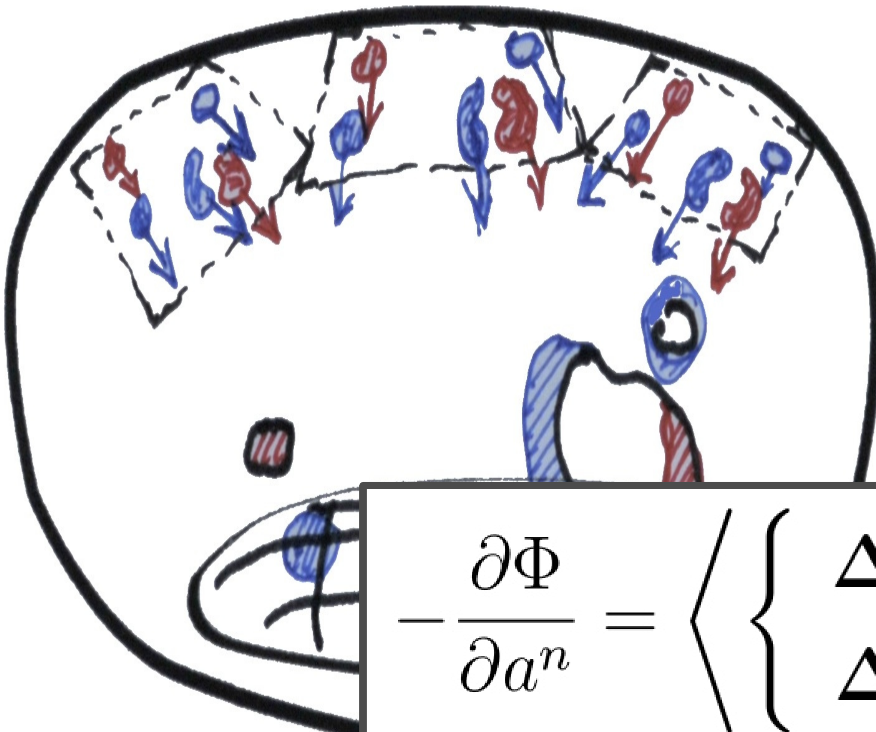
Compute the steepest descent of the misfit function.

- This is the right-hand-side of Newton's equations for the parameter adjustment.

$$-\frac{\partial \Phi}{\partial a^n} = \left\langle \mathcal{U}_{\boldsymbol{a}}^{*} \left\{ \begin{array}{c} \boldsymbol{\Delta p}_{\odot} \\ \boldsymbol{\Delta q}_{\odot} \end{array} \right\}, \left\{ \begin{array}{c} \boldsymbol{\Delta f}_{\odot}^{n} \\ \boldsymbol{\Delta g}_{\odot}^{n} \end{array} \right\} \right\rangle_{D\times\mathbb{S}^2 \oplus \Gamma_-}$$

Adjoint method:

compute the steepest descent of misfit



The adjoint Stokes-vector solution is on the correct 3D domain.

Compute the steepest descent of the misfit function.

$$-\frac{\partial \Phi}{\partial a^n} = \left\langle \left\{ \begin{array}{c} \boldsymbol{\Delta p}_\odot \\ \boldsymbol{\Delta q}_\odot \end{array} \right\}, \mathcal{U}_{\boldsymbol{a}} \left\{ \begin{array}{c} \boldsymbol{\Delta f}^n_\odot \\ \boldsymbol{\Delta g}^n_\odot \end{array} \right\} \right\rangle_{D\times\mathbb{S}^2\oplus\Gamma_+}$$

$$-\frac{\partial \Phi}{\partial a^n} = \left\langle \mathcal{U}^*_{\boldsymbol{a}} \left\{ \begin{array}{c} \boldsymbol{\Delta p}_\odot \\ \boldsymbol{\Delta q}_\odot \end{array} \right\}, \left\{ \begin{array}{c} \boldsymbol{\Delta f}^n_\odot \\ \boldsymbol{\Delta g}^n_\odot \end{array} \right\} \right\rangle_{D\times\mathbb{S}^2\oplus\Gamma_-}$$

How do we tell the computer?

computers know linear algebra.

Use the derivative:

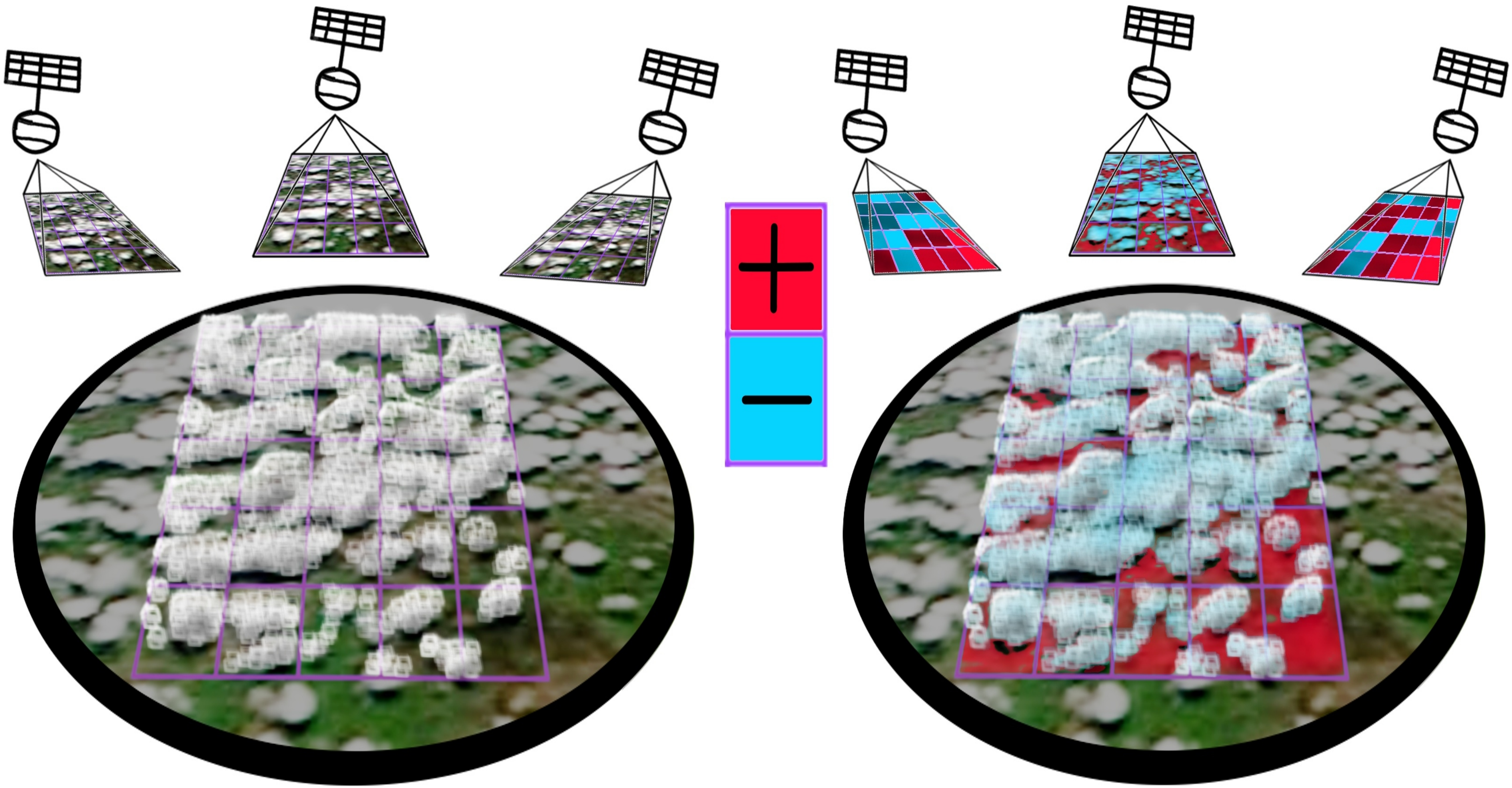
grad(Φ),

To setup the linear system:

Ax=b.

Adjoint method:

scalable adjustments to 3D properties



Adjoint method:
scalable adjustments to 3D properties

Iterative minimization of the misfit function with only two calls to the 3D VRTE (per wavelength):

- Solve the 3D VRTE once to compute the residual
- Solve the adjoint 3D VRTE once calculate the derivative
- Solve a system of linear equations for the parameter adjustment

Procedure scales to very large problems with . . .

- Many measurement constraints
- Many unknown cloud, aerosol and surface properties

Adjoint method makes 3D retrievals with the 3D VRTE worth discussing

- Future project 1: Test derivative calculations and performance
- Future project 2: Synthetic retrievals and inverse problem analysis

Adjoint methods for 3D
re

Thank you!!!

William Martin

Brian Cairns, Guillaume Bal

December 3, 2014

NASA Goddard Institute for Space Studies

Work supported by NASA Headquarters under the NASA Earth and Space Science Fellowship Program, Grant NNX-10AN85H, and in part by an IGERT grant from the US National Science Foundation to Columbia University.

Goddard Institute for Space Studies

Adjoint method:

The adjoint source from residuals

Recall the expression for each measurement:

$$y^m(\boldsymbol{a}) = \langle \boldsymbol{p}^m_\odot, \boldsymbol{u} \rangle_{D\times\mathbb{S}^2} + \langle \boldsymbol{q}^m_\odot, \boldsymbol{u} \rangle_{\Gamma_+}$$

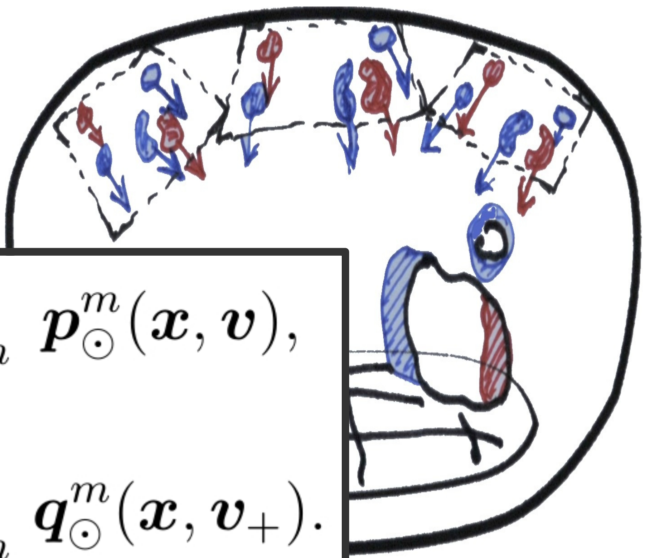
Avoid computing the Jacobian (sum over measurements first)

$$\begin{aligned} -\frac{\partial \Phi(\boldsymbol{a})}{\partial a^n} &= (\hat{\boldsymbol{y}} - \boldsymbol{y}(\boldsymbol{a}))^T \cdot (\boldsymbol{S}_\epsilon^{-1}) \cdot \frac{\partial \boldsymbol{y}(\boldsymbol{a})}{\partial a^n} \,. \\ &= \left\langle \boldsymbol{\Delta p}_\odot, \frac{\partial \boldsymbol{u}}{\partial a^n} \right\rangle_{D\times\mathbb{S}^2} + \left\langle \boldsymbol{\Delta q}_\odot, \frac{\partial \boldsymbol{u}}{\partial a^n} \right\rangle_{\Gamma_+} \end{aligned}$$

The adjoint source functions

$$\boldsymbol{\Delta p}_\odot(\boldsymbol{x}, \boldsymbol{v};\, \boldsymbol{a}) = \sum_{m'\,m} \left(\hat{y}^{m'} - y^{m'}(\boldsymbol{a})\right) \, \left(\boldsymbol{S}_\epsilon^{-1}\right)_{m'm} \, \boldsymbol{p}^m_\odot(\boldsymbol{x}, \boldsymbol{v}),$$

$$\boldsymbol{\Delta q}_\odot(\boldsymbol{x}, \boldsymbol{v}_+;\, \boldsymbol{a}) = \sum_{m'\,m} \left(\hat{y}^{m'} - y^{m'}(\boldsymbol{a})\right) \, \left(\boldsymbol{S}_\epsilon^{-1}\right)_{m'm} \, \boldsymbol{q}^m_\odot(\boldsymbol{x}, \boldsymbol{v}_+).$$



Adjoint method:
sources for parameter derivatives

Differentiate the VRTE:

$$\boxed{\frac{\partial}{\partial a^n}} \Rightarrow \boxed{\begin{aligned} \boldsymbol{v}\cdot\nabla\boldsymbol{u}+\sigma\boldsymbol{u}-\mathcal{Z}[\boldsymbol{u}] &= \boldsymbol{f}_{\odot} \\ \boldsymbol{u}|_{\Gamma_-}-\mathcal{R}\left[\boldsymbol{u}|_{\Gamma_+}\right] &= \boldsymbol{g}_{\odot} \end{aligned}}$$

3D VRTE with same left-hand-side:

$$\begin{aligned} \boldsymbol{v}\cdot\nabla\frac{\partial\boldsymbol{u}}{\partial a^n}+\sigma\frac{\partial\boldsymbol{u}}{\partial a^n}-\mathcal{Z}\left[\frac{\partial\boldsymbol{u}}{\partial a^n}\right] &= \boldsymbol{\Delta f}^n_{\odot}, \\ \left.\frac{\partial\boldsymbol{u}}{\partial a^n}\right|_{\Gamma_-}-\mathcal{R}\left[\left.\frac{\partial\boldsymbol{u}}{\partial a^n}\right|_{\Gamma_+}\right] &= \boldsymbol{\Delta g}^n_{\odot}, \end{aligned}$$

The derivative solves the 3D VRTE with the following right-hand-side source:

$$\begin{aligned} \boldsymbol{\Delta f}^n_{\odot}(\boldsymbol{x},\boldsymbol{v};\,\boldsymbol{a}) &= -\frac{\partial\sigma}{\partial a^n}(\boldsymbol{x};\boldsymbol{a})\;\boldsymbol{u}(\boldsymbol{x},\boldsymbol{v};\,\boldsymbol{a})+\frac{1}{4\pi}\int_{\mathbb{S}^2}\mathrm{d}S_{\boldsymbol{v}'}\;\frac{\partial\boldsymbol{Z}}{\partial a^n}(\boldsymbol{x},\boldsymbol{v},\boldsymbol{v}';\boldsymbol{a})\cdot\boldsymbol{u}(\boldsymbol{x},\boldsymbol{v}';\,\boldsymbol{a}) \\ \boldsymbol{\Delta g}^n_{\odot}(\boldsymbol{x},\boldsymbol{v}_-;\,\boldsymbol{a}) &= \frac{1}{2\pi}\int_{\boldsymbol{v}_+\cdot\nabla h(\boldsymbol{x})>0}\mathrm{d}S_{\boldsymbol{v}_+}\;\left|\boldsymbol{v}_+\cdot\nabla h(\boldsymbol{x})\right|\;\frac{\partial\boldsymbol{R}}{\partial a^n}(\boldsymbol{x},\boldsymbol{v}_-,\boldsymbol{v}_+;\boldsymbol{a})\cdot\boldsymbol{u}(\boldsymbol{x},\boldsymbol{v}_+;\,\boldsymbol{a}). \end{aligned}$$

Adjoint method:
sources for parameter derivatives

$$\begin{Bmatrix} \dfrac{\partial \boldsymbol{u}}{\partial a^n}\Big|_{D\times\mathbb{S}^2} \\ \dfrac{\partial \boldsymbol{u}}{\partial a^n}\Big|_{\Gamma_+} \end{Bmatrix} = \mathcal{U}_{\boldsymbol{a}} \begin{Bmatrix} \boldsymbol{\Delta f}^n_{\odot} \\ \boldsymbol{\Delta g}^n_{\odot} \end{Bmatrix}$$

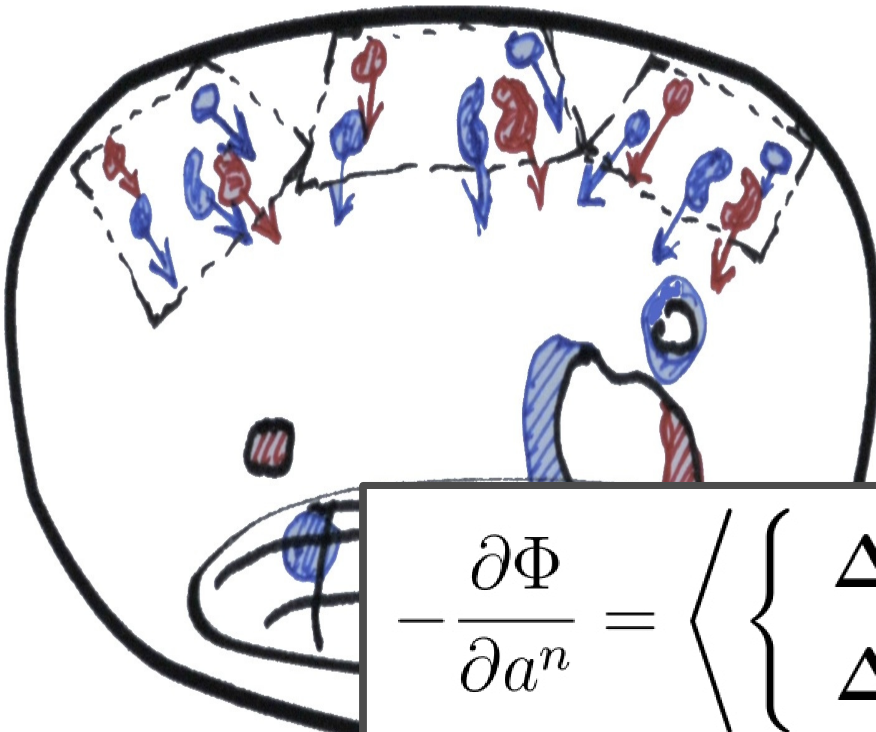
The derivative solves the 3D VRTE with the following right-hand-side source:

$$\begin{aligned} \boldsymbol{v}\cdot\nabla\frac{\partial \boldsymbol{u}}{\partial a^n} + \sigma\frac{\partial \boldsymbol{u}}{\partial a^n} - \mathcal{Z}\left[\frac{\partial \boldsymbol{u}}{\partial a^n}\right] &= \boldsymbol{\Delta f}^n_{\odot}, \\ \frac{\partial \boldsymbol{u}}{\partial a^n}\Big|_{\Gamma_-} - \mathcal{R}\left[\frac{\partial \boldsymbol{u}}{\partial a^n}\Big|_{\Gamma_+}\right] &= \boldsymbol{\Delta g}^n_{\odot}, \end{aligned}$$

$$\boldsymbol{\Delta f}^n_{\odot}(\boldsymbol{x},\boldsymbol{v};\,\boldsymbol{a}) = -\frac{\partial\sigma}{\partial a^n}(\boldsymbol{x};\boldsymbol{a})\;\boldsymbol{u}(\boldsymbol{x},\boldsymbol{v};\,\boldsymbol{a}) + \frac{1}{4\pi}\int_{\mathbb{S}^2}\mathrm{d}S_{\boldsymbol{v}'}\;\frac{\partial \boldsymbol{Z}}{\partial a^n}(\boldsymbol{x},\boldsymbol{v},\boldsymbol{v}';\boldsymbol{a})\cdot\boldsymbol{u}(\boldsymbol{x},\boldsymbol{v}';\,\boldsymbol{a})$$

$$\boldsymbol{\Delta g}^n_{\odot}(\boldsymbol{x},\boldsymbol{v}_-;\,\boldsymbol{a}) = \frac{1}{2\pi}\int_{\boldsymbol{v}_+\cdot\nabla h(\boldsymbol{x})>0}\mathrm{d}S_{\boldsymbol{v}_+}\;\left|\boldsymbol{v}_+\cdot\nabla h(\boldsymbol{x})\right|\frac{\partial \boldsymbol{R}}{\partial a^n}(\boldsymbol{x},\boldsymbol{v}_-,\boldsymbol{v}_+;\boldsymbol{a})\cdot\boldsymbol{u}(\boldsymbol{x},\boldsymbol{v}_+;\,\boldsymbol{a}).$$

Adjoint method:
compute the steepest descent of misfit



The adjoint Stokes-vector solution is on the correct 3D domain.

Compute the steepest descent of the misfit function.

$$-\frac{\partial \Phi}{\partial a^n} = \left\langle \left\{ \begin{array}{c} \boldsymbol{\Delta p}_{\odot} \\ \boldsymbol{\Delta q}_{\odot} \end{array} \right\}, \mathcal{U}_{\boldsymbol{a}} \left\{ \begin{array}{c} \boldsymbol{\Delta f}^n_{\odot} \\ \boldsymbol{\Delta g}^n_{\odot} \end{array} \right\} \right\rangle_{D\times\mathbb{S}^2 \oplus \Gamma_+}$$

$$-\frac{\partial \Phi}{\partial a^n} = \left\langle \mathcal{U}^*_{\boldsymbol{a}} \left\{ \begin{array}{c} \boldsymbol{\Delta p}_{\odot} \\ \boldsymbol{\Delta q}_{\odot} \end{array} \right\}, \left\{ \begin{array}{c} \boldsymbol{\Delta f}^n_{\odot} \\ \boldsymbol{\Delta g}^n_{\odot} \end{array} \right\} \right\rangle_{D\times\mathbb{S}^2 \oplus \Gamma_-}$$

How do we tell the computer?

computers know linear algebra.

Adjust parameters with a step, b, which solves approximate Newton's equations:

$$\left(\nabla\nabla\Phi(\boldsymbol{a}) + \nabla\nabla\Phi_{\mathrm{prior}}(\boldsymbol{a})\right)\cdot\boldsymbol{b} = -\left(\nabla\Phi(\boldsymbol{a}) + \nabla\Phi_{\mathrm{prior}}(\boldsymbol{a})\right)$$

Approximate the second derivative with the gradient (Broyden-Fletcher-Goldfarb-Shanno):

$$\nabla\nabla\Phi(\boldsymbol{a}_k) \approx \boldsymbol{H}_k\left(\boldsymbol{a}_0, \cdots, \boldsymbol{a}_k, \nabla\Phi(\boldsymbol{a}_0), \cdots, \nabla\Phi(\boldsymbol{a}_k)\right)$$